Intellectica, 2006/2, **44**, pp. 121-143

Faciliter les activités des utilisateurs d'environnements informatiques : quoi, quand, comment ?

Alain MILLE[*], Guy CAPLAT[#], Mick PHILIPPON[§]

RESUME. Aider un utilisateur dans ses activités médiées par un environnement informatique pose des questions liées à la complexité potentielle du couplage utilisateur-environnement. Une combinatoire de situations imprévisibles rend toute tentative d'aide prescriptive très difficile et le terme faciliter sera préféré au terme assister. Ce sont les traces d'interaction, considérées comme inscriptions de l'expérience d'utilisation tracée, qui sont proposés comme source de facilitation en situation. L'hypothèse d'une empathie entre la trace inscrivant l'expérience et l'expérience vécue par l'utilisateur est à la base du principe de facilitation proposé. Cette hypothèse repose sur la capacité pour l'utilisateur de s'approprier le dispositif de facilitation en lui donnant les moyens de négocier la manière dont ce dispositif interviendra dans le processus de facilitation. Une réalisation sous forme d'un compagnon développé pour faciliter l'appropriation d'une suite logicielle de conception est détaillée pour illustrer de manière concrète les principes et les propositions faites. La discussion ouvre la voie à la formalisation de l'approche de facilitation à partir de traces en proposant de considérer les traces d'utilisation comme de nouveaux objets informatiques à la base d'une exploitation métacognitive explicite par l'utilisateur.

Mots clés : Expérience tracée, traces informatiques, métacognition, cognition située, système de facilitation.

ABSTRACT. Facilitating Computer Based User Activities: What, When, How? Helping a user through her computer mediated activities arise important questions related to the complexity of the user-environment coupling. Unpredictable combinative situations make unrealistic attempts to help users in a prescriptive way. The term facilitating will be preferred to the term assisting for this reason. In this paper, we consider interaction traces as traced use experience inscriptions which are exploited as situated facilitating sources. The hypothesis of a possible empathy between these inscriptions and the lived experience of the user founds the facilitating process we propose. This hypothesis lies on the ability for the user to appropriate the facilitating process because of the possibility for her to negotiate the way facilitating is processed. We illustrate the proposal by a companion which has been developed in order to facilitate the appropriation of a CAO package. This illustration shows how, concretely, traces are exploited as facilitating sources. Discussion draws a roadmap about traces systems, i.e., formalization of traces as new computer objects founding meta-cognitive exploitation by computer environment users.

Key words: traced experience, computer traces, meta-cognition, situated cognition, helping system.

[*] LIRIS CNRS UMR 5205, U Lyon1, INSA Lyon, U Lyon2, ECL, alain.mille@liris.cnrs.fr.
[#] INSA Lyon, Département Informatique guy.caplat@insa-lyon.fr.
[§] Bourse Cifre, Dassault Systèmes, Suresnes, France, mick.philippon@laposte.net.

© 2006 Association pour la Recherche Cognitive.

1. INTRODUCTION

Le titre de l'article impose une clarification du sujet dont il sera question dans ce papier. *Faciliter* ne signifie pas exactement *Assister* : nous considérons la notion de facilitation comme moins exigeante en matière de modélisation de l'activité d'un utilisateur que l'assistance qui laisserait supposer une certaine *compétence* de l'environnement en la matière. *Faciliter* nécessite une boucle d'*inter-action* avec l'utilisateur seul juge de l'effet de facilitation pour lui. Dans la suite de l'article nous utiliserons le terme générique *aide* quand il s'agira de désigner sans les distinguer les processus de facilitation ou d'assistance. L'*activité* est considérée dans l'acception usuelle du terme comme *ensemble des actions et des opérations effectuées (dans un domaine particulier)*. La notion d'environnement informatique est préférée à application, trop restrictif, ou système, trop englobant peut-être. Cette notion d'environnement informatique est utilisée sans qu'elle puisse être précisée vraiment. Pour faciliter l'exposé des principes que nous allons défendre ici, nous ferons l'hypothèse d'un environnement informatique au contour défini par une *suite logicielle* particulière. Nous appelons *suite logicielle*, un ensemble de logiciels proposés de manière cohérente autour d'une activité générique. Tout le monde connaît les « suites bureautiques », il sera ici question d'une « suite de conception » pour illustrer le propos, en l'occurrence la suite CATIA[1] V5. Malgré la limitation du périmètre, il reste que l'usage d'un tel environnement par des utilisateurs variés dans des contextes différents révèle une complexité et une dynamique très fortes. En effet, les services et fonctions disponibles sont très nombreux, se présentent sous des formes multiples et peuvent être associés de manière combinatoire pour des tâches jamais complètement prescrites. L'appropriation de ces environnements pour réaliser des tâches spécifiques à partir des outils génériques est une question difficile, d'autant plus que les tâches en question sont souvent redéfinies durant leur réalisation. Un « mode d'emploi » d'un tel environnement est non seulement une entreprise particulièrement difficile, comme l'explique bien Dominique Boullier dans ce même numéro, mais exigera de toutes façons une implication de l'utilisateur pour qu'il soit utile. Le hiatus entre concepteur(s)-auteur(s) et utilisateur(s)-lecteur(s) est bien entendu déjà pris en compte par les méthodes de conception « centrées utilisateurs », mais dans le cas de « suites logicielles » comme nous avons choisi de les nommer ici, la variabilité des usages est telle que seule une aide fonctionnelle de l'environnement peut être réellement concernée par de tels manuels. En effet, les usages *singuliers* échappent par définition à la chaîne de conception et de re-conception. Notre objet de recherche, principalement illustré dans cet article par une réalisation empirique, est précisément la prise en compte de la *singularité* au moment de la situation d'usage, dans le cadre d'une activité générique connue (la conception avec CATIA), mais pour des tâches non explicitement définies (non décrites dans une bibliothèque CATIA). Plus précisément, nous nous intéresserons à proposer une méthode générale permettant de proposer un contenu facilitant (le *QUOI)* en se fondant sur les inter-actions de l'utilisateur. Nous écrivons *inter-action* plutôt *qu'interaction* pour dénoter le fait que les

[1] CATIA est un produit Dassault Systèmes, support de l'ensemble du processus de conception, de l'avant-projet à la maintenance, en passant par la conception détaillée, la simulation et l'assemblage http://www.3ds.com/fr/corporate/about-us/brands/catia/

actions de l'utilisateur et les réactions de l'environnement sont dépendantes les unes des autres dans le cadre d'une activité particulière. L'utilisateur agit avec l'environnement comme inter-médiaire avec lui-même d'abord. Il n'est pas possible de dissocier l'action de l'attente du résultat de l'action qui précisément est le lieu de la confrontation *logiques* d'utilisation / *logiques* de conception. Le terme *logique* indique que nous faisons l'hypothèse de rationalité dans l'utilisation comme dans la conception et que ces *logiques* sont différentes par définition puisque les concepteurs et les utilisateurs ont des objectifs distincts mais non indépendants.

L'article se centre sur les aspects empiriques de la démarche dans un contexte technique particulier et s'attache tout d'abord à rappeler rapidement la genèse de ce travail au sein de notre équipe et il n'évoquera que succinctement les résonances repérées avec des théories issues d'autres disciplines, en particulier en Sciences Humaines et Sociales. Une *modélisation de la trace* pour rendre compte de l'expérience d'utilisation d'un environnement informatique sera ensuite présentée avant de préciser la notion de facilitation telle que nous l'utilisons dans notre approche. Le cœur de l'article sera donc dédié à la présentation d'une réalisation particulière sous la forme d'un *compagnon* pour CATIA montrant une implémentation du modèle théorique permettant de répondre d'une certaine façon au QUOI et au QUAND du titre. Cette réalisation se focalise sur une exploitation de la trace comme source de connaissances pour faciliter une activité complexe et présente une manière originale d'ajouter le QUAND au QUOI issu de la théorie. La discussion sera l'occasion de jeter un œil critique sur la réalisation empirique proposée et argumentera sur les nécessaires croisements disciplinaires pour avancer dans l'élaboration d'environnements informatiques intégrant les principes de facilitation d'appropriation que nous défendons. La conclusion devrait permettre de synthétiser les apports et les limites de notre contribution au difficile problème de l'appropriation des environnements techniques complexes.

2. GENESE ET RESONANCES THEORIQUES DE L'APPROCHE

2.1 Genèse

La genèse de notre démarche est jalonnée de différents travaux empiriques autour de la question de l'aide à l'utilisateur d'un environnement informatique pour des tâches qui sont peu explicitées mais pour lesquelles il est possible de disposer d'une collection d'expériences concrètes. La collection d'expériences induit la description – et par là la modélisation – de ces expériences concrètes. C'est assez naturellement que nous nous sommes intéressés au Raisonnement à Partir de Cas (RàPC)(Aamodt et Plaza, 1994; Riesbeck et Schank, 1989; Schank, 1982) comme première réponse à cette exigence. D'emblée, pourtant, nous avons envisagé cet usage non pas comme un système expert excluant l'utilisateur mais bien comme un système d'aide à la tâche de l'utilisateur. Le Raisonnement à Partir de Cas (RàPC), tel que classiquement traité par la communauté informatique s'inscrit classiquement dans une catégorie d'ambition forte avec l'idée principale de diminuer l'effort lié à l'acquisition des connaissances, s'inspirant des principes issus de travaux en psychologie (Schank, 1982) sans échapper toutefois à les réduire à une méthode efficace de résolution de problèmes. L'ambition formelle du calcul est moins grande que dans les approches logiques puisqu'il s'agit essentiellement d'un raisonnement analogique de nature abductive : un épisode de résolution de problème est constitué

d'une partie description de problème et d'une partie description de solution qu'une fois associées, on appelle un cas ; un nouvel épisode de résolution représenté par sa partie problème est résolu par adaptation de la solution d'un épisode passé retrouvé sur la base de la similarité mesurée entre les parties problèmes des épisodes ; il s'agit donc d'une hypothèse qui sera confrontée à l'environnement ; la confrontation permet de réviser et corriger la solution ; le nouvel épisode est organisé dans la mémoire des épisodes passés. Une autre originalité de cette approche, qui en a fait probablement le succès pour des applications assez nombreuses, est donc que le cycle de raisonnement permet d'intégrer de nouveaux cas de manière relativement simple, offrant ainsi les apparences d'un système qui apprend de l'expérience de son utilisation. Les travaux sur le RàPC (Champin, 2003; Corvaisier, et al. 1997; Corvaisier, et al., 2000; Héraud et Mille, 2003; Herbeaux et Mille, 1999; Mille, et al., 1999) que nous avons menés nous montrent que l'utilisateur a tendance à vouloir également utiliser les cas pour d'autres problèmes que ceux pour lesquels le système a été prévu. Autrement dit, l'épisode de résolution normalement décomposé en une partie problème et une partie solution, clairement prédéfinies, est également vu selon d'autres décompositions problème-solution, par une subversion bien classique des environnements informatiques par leurs utilisateurs. Dans ce type de subversion, l'utilisateur n'est alors pas bien aidé par le système initialement mis en place si ce n'est pour lui permettre de spécifier directement ou indirectement une nouvelle mesure de similarité lui permettant de retrouver... ce qu'il cherche! Les cas se révèlent alors eux mêmes comme des inscriptions documentaires de connaissances bien utiles pour l'utilisateur. C'est à l'occasion d'une application d'aide à la supervision industrielle (Mille, et al., 1999) que les limites du Raisonnement à Partir de Cas *standard* se sont révélées les plus gênantes. En effet, il n'était pas possible de savoir à l'avance quelles seraient les *limites* d'un épisode de supervision sans le concours de l'utilisateur. Nous avions alors exploité quelques heuristiques sur la *fenêtre temporelle* de prise en compte des événements pertinents pour un épisode en cours (un cas cible) pour y repérer des *objets de supervision* pertinents qui s'ajoutaient à ceux que l'examen de la supervision courante (ce qui est présent sur les écrans) fournissait. Un *objet focal* désigné par l'utilisateur tentait de prendre en compte le *point de vue* de l'utilisateur. Nous avions alors défini une façon générique de considérer un contexte de supervision à prendre en compte. Bien entendu, si ces heuristiques s'avéraient inefficaces, il était impossible pour l'utilisateur d'y accéder afin de les modifier même de manière indirecte. La conclusion assez évidente est qu'il est vain de vouloir *prévoir à l'avance* ce qui sera *contexte pertinent* pour une activité d'un utilisateur sans interagir avec lui. Nous avons alors travaillé sur une forme de généralisation du raisonnement à partir de cas, en repoussant au moment de la mise en situation la définition d'un contexte. Pour y parvenir, ce ne sont plus des épisodes particuliers de résolution de problème qui sont enregistrés, mais *une trace d'utilisation*, conteneur potentiel d'épisodes qui seront alors situés temporellement et conceptuellement par les relations que les objets d'intérêt tracés entretiennent les uns avec les autres. L'idée générale est d'exploiter cette trace avec des requêtes issues directement ou indirectement de l'interaction avec l'utilisateur pour y déterminer des épisodes passés potentiellement utiles dans la situation courante.

2.2 RESONANCES THEORIQUES D'UNE APPROCHE D'EXPERIENCE TRACEE

Nous nous contenterons dans cette partie de relever les théories avec lesquelles l'approche que nous proposons semble résonner. L'acception du terme *expérience* que nous retenons nous fait considérer *l'expérience comme interaction* comme certains commentateurs de l'œuvre de Dewey (Dewey, 1934) le comprennent dans son œuvre, première résonance théorique que nous relevons. Toute connaissance y trouverait sa source de manière plus ou moins directe. Raisonner consisterait alors à associer les expériences préalablement établies pour révéler la connaissance dans le cadre d'une nouvelle expérience. D'un point de vue *Ingénierie de la connaissance*, les connaissances révélées peuvent être synthétisées en *expertise*, sous la forme d'inscriptions de connaissances documentées et accompagnées de leur modèle d'interprétation autorisant l'inférence calculée. L'expérience sensible étant par définition non accessible en tant que telle puisque incorporée, au sens propre du terme, seules les traces laissées dans l'environnement (y compris probablement chez le sujet vivant l'expérience) restent des inscriptions d'une connaissance émergente ou/et mobilisée au moment d'une nouvelle expérience. Ces inscriptions dans l'environnement permettent de faciliter à nouveau l'émergence du sens dans de nouvelles expériences dont elles composent également d'une manière ou d'une autre l'environnement. Cette notion d'inscription de la connaissance a été pointée et argumentée par Bruno Bachimont (Bachimont, 2004) qui propose le concept d'ingénierie des inscriptions de la connaissance dans le cadre d'expertises documentées, et résonne aussi aux réflexions de Bernard Stiegler sur les notions de rétentions primaires, secondaires et tertiaires, ces dernières rendant « possible l'extériorisation de l'expérience individuelle dans des traces, et comme transmission. » (Stiegler, 2005, page 23, alinéas 10-13). Dans le cadre restreint des systèmes informatiques, les inscriptions de la connaissance peuvent être le résultat d'un processus volontaire d'ingénierie de la connaissance (Charlet, 2004) cherchant à garder trace de la connaissance sous une forme la plus exploitable possible par un environnement informatique, c'est à dire en cherchant à en représenter la sémantique par des calculs dont on pourra penser qu'ils fournissent à la demande l'interprétation convenable correspondant à la connaissance inscrite.

Pour notre part, nous nous intéressons tout particulièrement au traçage explicité à l'utilisateur lui permettant de *se voir agir* d'une manière ou d'une autre, et lui permettant d'utiliser cette trace de son expérience comme support pour faciliter ses activités ou pour construire un discours argumenté en vue d'échange ou de partage par exemple. Cette trace serait alors une *inscription de l'expérience tracée* nécessitant des modèles d'interprétation dynamiques et plastiques en inter-action avec l'utilisateur générateur conscient de ces traces. Le pari est que cette inscription convenablement médiée enrichisse l'environnement de manière utile au processus de cognition située. Cette approche résonne avec les travaux de Christian Brassac (Brassac, 2006), ceux de Pierre Rabardel (Rabardel, 1995) et plus généralement encore avec les courants de pensée fondés par Vygotski (Clot, 1999) ou Leontiev (Leontiev, 1984). Ces différents travaux montrent que l'utilisateur et l'artefact technique qu'il utilise sont co-constructeurs du sens pendant l'activité, en situation. Nous tentons de faciliter cette construction du sens en fournissant à l'utilisateur un *terrain* de confrontation entre les logiques de conception de l'environnement et ses logiques d'utilisation pendant l'activité. L'environnement informatique constitue un terrain d'inter-actions où se confrontent ces logiques, et nous proposons

donc de matérialiser de manière spécifique cette confrontation par une *trace d'utilisation*, autorisant ainsi une prise de conscience plus aisée par l'utilisateur de ses inter-actions et des moyens d'exploitation de cette trace pour faciliter son activité par la réutilisation en situation de son expérience, mais aussi pour faciliter l'exportation et le partage de son expérience ainsi tracée.

3. UNE MODELISATION DE LA TRACE D'UTILISATION

Nous présentons d'abord le principe de modélisation de l'utilisation en introduisant les notions de *modèle d'utilisation (MU)* et de *signatures de tâches expliquées* (SiTEx) avant de détailler une manière de les mettre en œuvre avec l'approche MUSETTE.

3.1 Modèle d'utilisation et signatures de tâches expliquées

Nous appelons *modèle d'utilisation* l'ensemble des éléments qui permettent de produire une trace d'utilisation sur la base des inter-actions avec un environnement informatique. La construction d'un modèle d'utilisation nécessite qu'un modélisateur humain observe des utilisateurs dans leurs pratiques de l'environnement. L'observation « initiale » est réalisée dans le cadre d'un environnement informatique *S* ne comportant pas d'objets traces. C'est une observation « ouverte »[2], destinée à identifier ce que l'utilisateur prend en considération comme entités et événements lorsqu'il inter-agit avec l'environnement.

Une façon concrète de construire un modèle d'utilisation initial est de mettre en évidence ce que nous avons convenu d'appeler des « signatures de tâches expliquées » (SiTEx). Une SiTEx est un motif inter-actionnel que les utilisateurs re-connaissent dans une trace comme typique d'une tâche qu'ils peuvent alors expliquer (décrire). Les signatures peuvent avoir des « grains » variés, depuis une séquence d'opérations élémentaires jusqu'à une combinaison de possibilités d'inter-actions offertes par différents environnements, considérés comme un seul environnement. La manière d'exprimer les signatures fournit matière à symboliser les processus inter-actionnels pour mieux les manipuler ensuite. Le modèle d'utilisation initial doit permettre de produire ce que nous avons convenu d'appeler une « trace primitive ». Celle-ci constitue le premier niveau considéré comme interprétable par l'utilisateur d'une trace et exploitable par la machine via les SiTEx permettant de repérer des séquences inter-actionnelles potentiellement signifiantes.

Le modèle d'utilisation d'un environnement[3] *S* est utilisé pour mettre en place l'objet informatique *Trace* qui permet donc de « parler » de ce que l'on fait en inter-agissant avec *S*.

---

[2] Par opposition à une observation « fermée » dans laquelle on ne ferait que vérifier des choses. Il est nécessaire d'observer les situations d'utilisations en tant que telles, sans vouloir les découper *a priori* selon une vision qui ne pourrait venir que de pré-conçus.

[3] Il faut remarquer que le modèle d'utilisation d'un environnement est une façon de donner le contour *perçu* de l'environnement informatique. Le contour de l'environnement *conçu* n'est en général pas accessible.

3.2 MUSETTE : Modéliser les Utilisations et les Tâches pour Tracer l'Expérience

L'approche Musette (Champin, et al., 2003) a été mise en place dans l'objectif de fournir un cadre pour l'aide à base de trace au sein de l'équipe CEXAS[4]. La Figure 1 illustre l'approche MUSETTE, articulant le niveau du traçage (en haut à droite) et celui de la réutilisation d'épisodes d'utilisation (à gauche). Le schéma se lit de la façon suivante : un utilisateur inter-agit avec un environnement (système). Un agent observateur, guidé par un modèle d'observation, génère à partir de ces inter-actions une trace primitive respectant un modèle d'utilisation. Un analyseur générique de trace en extrait alors des épisodes, en accord avec des signatures de tâches expliquées. Ces épisodes sont réutilisés par des agents assistants, dont l'action peut être soit spécifique avec des agents clairement distincts du système, soit intégrée grâce au système lui-même. Musette est une approche de modélisation orientée « processus » d'utilisation, et peut être implémentée grâce à divers langages ou formalismes de représentation, à même d'en représenter les différents éléments, que nous précisons dans ce qui suit[5]. Il convient de déterminer de quoi la trace sera constituée (modèle d'utilisation), et comment celle-ci sera générée (modèle d'observation). Les éléments d'inter-action constituants de la trace sont des *objets d'intérêt* (OI) de trois types : *entités*, *évènements* et *relations*. Les entités sont des objets « présents » pour l'utilisateur dans son inter-action ; les évènements des objets qui « ont lieu », qui « se passent », durant cette même inter-action ; les relations peuvent lier indifféremment des entités et/ou des évènements.

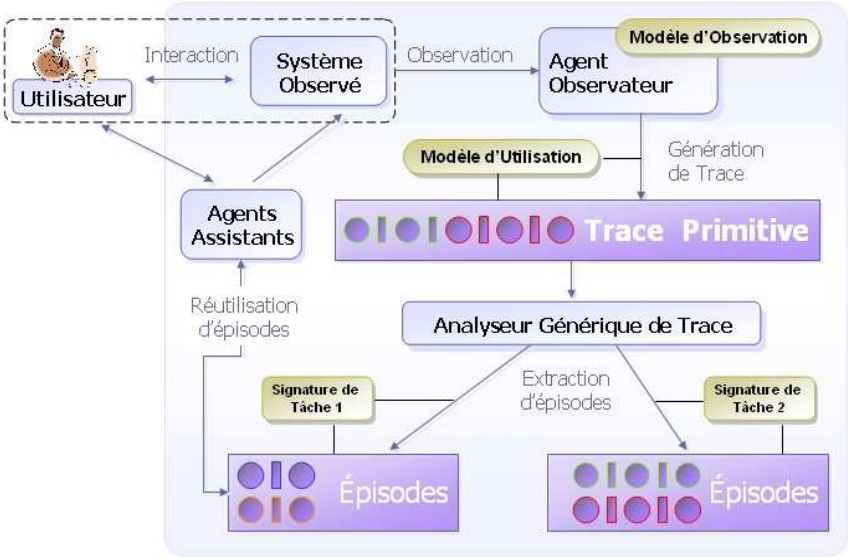


Figure 1. Approche Musette

[4] Cognition, EXpérience et Agents Situés, équipe de recherche du LIRIS.

[5] Les contraintes éditoriales nous contraignant à aller à l'essentiel, nous invitons le lecteur intéressé à consulter les références citées, qui discutent plus précisément de nombreux points ici survolés ou passés sous silence.

Un modèle d'utilisation décrit les entités, évènements et relations à observer pour construire une trace primitive, à la manière d'une « ontologie de l'observation ». Associé au MU, un modèle d'observation décrit les règles nécessaires à la détermination des données pertinentes issues de l'environnement et à la construction effective de la trace primitive. Le modèle d'observation n'est pas spécifié dans l'approche Musette et doit être adapté à chaque environnement observé.

La trace primitive est structurée selon deux types de structures : les états, regroupant des entités, et les transitions regroupant des évènements. Les premiers représentent l'état du système observé à un instant/période donné par rapport au modèle d'utilisation. Une transition regroupe un ensemble d'évènements, observés entre deux états. La trace est donc *in fine* une séquence d'états et de transitions[6].

La Figure 2 présente un exemple simplifié d'un modèle d'utilisation de navigateur Web, associé à un fragment de trace primitive. Les entités du modèle d'utilisation sont les pages Web *(Page)*, les Hyperliens *(Lien*), les images *(Img)* et les préférences *(Pref)*. Les évènements du modèle d'utilisation sont les clics de l'utilisateur *(Clic)*, ses pages sauvegardées localement *(Sauv*), ses bookmarks sur une page *(Bm*) et le changement de langue *(Lang)*. Dans la suite de cet exemple, nous considèrerons que les pages possèdent également un attribut URL contenant leur adresse.

Pour la lisibilité de la figure, toutes les relations possibles du MU n'y ont pas été reportées. Seules apparaissent les relations d'une entité (*Lien*) vers un évènement (*Clic*), et celle d'un évènement (*Clic*) vers une entité (*Page*) - signifiant simplement que le lien cliqué mène à la nouvelle page affichée. D'autres relations apparaissent uniquement dans le fragment de trace, mais sont en principe définies dans le modèle d'utilisation : d'une page à un lien (« la page contient le lien »), d'une page à un bookmark (« le bookmark s'applique à cette page »), de page à page (« cette page est rafraîchie »), de préférence à changement de langage, puis à une autre préférence (quand les préférences représentent les choix de l'ancien et du nouveau langage). On peut noter que la transition 6 comporte deux évènements, indépendants l'un de l'autre dont la trace ne donne pas l'ordre d'apparition. La relation (*Persistance*), entre l'entité (*fr*) de l'(*état 5*) et l'entité (*fr*) de l'(*état 6*), permet quant à elle, dans ce modèle d'utilisation particulier, de traduire le fait que c'est un même objet du point de vue du système, qui se retrouve dans deux états successifs.

[6] Le choix des termes *État* et *Transition* est lié à la volonté de considérer le cours d'action comme un processus.

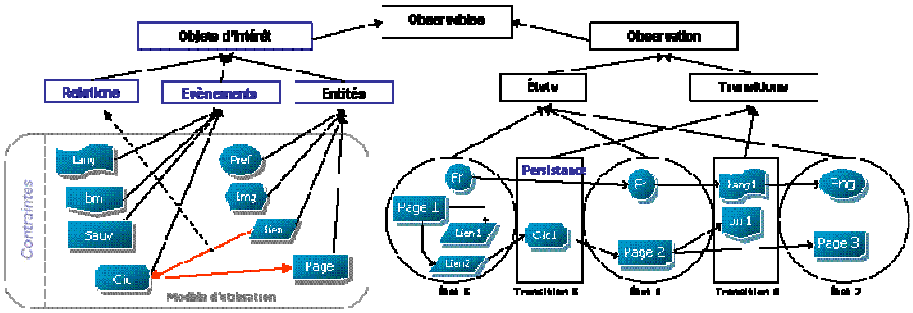


Figure 2. Un modèle d'utilisation simplifié et un fragment de trace de navigation

Nous appelons épisode une sous-partie de la trace primitive, qui peut y être découpée à l'aide d'une signature de tâche expliquée. Une telle signature s'exprime par : (a) des motifs de graphe constitués d'objets d'intérêt et leurs relations, (b) des contraintes relatives aux positions (distances relatives) des OIs dans la trace, (c) des contraintes sur la structure interne des objets d'intérêt. Les « explications » d'une signature de tâche permettent d'apporter des précisions sur les épisodes qu'elle a détectés : un épisode sera au minimum nommé, éventuellement annoté librement – en vue d'une lecture par l'utilisateur – ou formellement – en vue d'une utilisation automatique.

Nous présentons ci-dessous deux tâches simples de navigation Web qui peuvent être identifiées par notre modèle d'utilisation simplifié, et montrons comment leurs signatures peuvent être expliquées par des annotations sur les objets d'intérêt correspondants. Premier exemple : lorsqu'un utilisateur trouve une page intéressante, il lui arrive souvent de vouloir poser un bookmark sur le site dans lequel il a trouvé la page, considérant qu'il en contient – ou contiendra – d'autres. Ce genre de tâche est aisément repérable dans la trace primitive (fig. 3) : depuis une page (*Page*) d'un site, l'utilisateur remonte à la page d'accueil et y pose un bookmark (*Bm*). Des explications sont également fournies dans cette SiTEx : l'évènement bookmark (*Bm*) est annoté avec une explication textuelle, potentiellement utile pour l'interprétation de l'épisode et les deux pages Web impliquées peuvent être annotées par « page interne » et « page de garde », selon une ontologie des sites Web, définie en dehors de Musette, et utilisable par un assistant dédié.

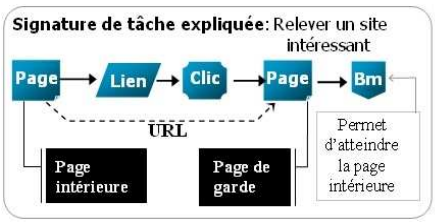


Figure 3. SiTEx et épisode (exemple 1)

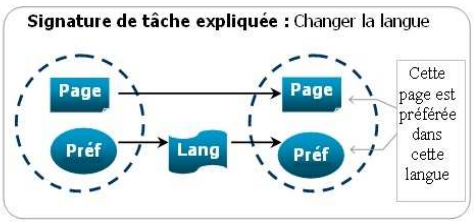


Figure 4. SiTEx et épisode (exemple 2).

Une autre tâche consiste à modifier les préférences de langue pour visualiser différentes versions de pages Web. Le changement étant représenté par un évènement Lang (lorsque la page change sans avoir cliqué sur un lien). La SiTEx de la figure 4, exprime une signature de ce type, les cercles en pointillés représentant une contrainte de co-occurrence des objets d'intérêt (*Page*) et (*Pref*) (représentant le langage courant de l'utilisation). Dans cet exemple des

annotations textuelles donnent des explications quant aux objets d'intérêt concernés.

4. FACILITATION / AIDE : DEFINITIONS

Nous avons choisi d'appeler « facilitateur » (Laflaquière et Ciaccia, 2005) tout agent informatique exploitant le contenu de la trace d'utilisation pour fournir à l'utilisateur des éléments de compréhension de l'environnement par rapport à ses propres utilisations et la capacité de s'approprier les possibilités d'inter-action offertes dans le cadre de ses utilisations propres. Le facilitateur doit donc « suivre » au plus près le contexte d'utilisation, défini par l'utilisateur lui-même en train d'agir. Une requête de facilitation consiste pour l'utilisateur à fournir, directement ou non, un contexte d'utilisation qui fait sens pour lui et pour lequel il espère trouver des éléments complémentaires élargissant sa compréhension de la situation. Il utilise le mécanisme de signature de tâche expliquée comme support de sa requête. La signature peut être fournie de différentes façons selon le degré d'anticipation de la facilitation :

- *Nouvelle signature de tâche expliquée* : l'utilisateur réalise des interactions, exploite la trace pour choisir les éléments qui font sens dans sa tâche courante et utilise les éléments choisis comme constitutifs de la signature de tâche expliquée qui sera sa requête. Le facilitateur recherche dans la trace les séquences « similaires » et les propose comme épisodes candidats en les situant temporellement et par les objets qui étaient alors en relation, ce qui permet à l'utilisateur d'élargir son interprétation et d'agir en conséquence.
- *Signature de tâche expliquée cataloguée* : l'utilisateur choisit un squelette de requête dans une bibliothèque de signatures de tâches expliquées. Celle qui correspond à son utilisation courante (qui sert d'index) et les épisodes correspondants sont retrouvés par l'agent facilitateur comme dans le premier cas. Les écarts constatés entre les épisodes retrouvés et l'épisode en cours de construction sont mis en évidence pour faciliter l'adaptation par l'utilisateur lui-même.
- *Signature de tâche expliquée canonique* : un modélisateur de la trace a conçu (en liaison avec des utilisateurs) des signatures canoniques à respecter, ainsi qu'un agent de type assistant permettant la mise en œuvre d'une utilisation conforme aux signatures en exhibant des épisodes réels qui pourront être adaptés. Cet usage des signatures de tâche expliquée revient à faire du Raisonnement à Partir de Cas à des fins d'aide. La partie suivante illustre l'utilisation de cette dernière approche pour la réalisation d'un agent d'aide dans le domaine de l'apprentissage humain, en se plaçant dans le dernier cas.

5. LE COMPAGNON CATIA

Cadre applicatif du compagnon

Ce développement prend place dans le cadre de la réalisation d'un assistant pour CATIA, le logiciel de CFAO de Dassault Systèmes qui permet la modélisation d'objets extrêmement variés, d'une simple vis à des assemblages très complexes comme des bateaux, des avions, des usines, … La Figure 5 présente un écran de CATIA, permettant de se rendre compte de la complexité du logiciel. CATIA est organisé en ateliers, qui regroupent des commandes ayant une

sémantique commune. Ainsi, on trouvera les ateliers de construction navale, de modélisation d'usine, de modélisation de surfaces, de simulation thermique... Plus de 150 ateliers cohabitent ainsi, regroupant environ 2000 commandes. La création d'un système d'aide pour un logiciel d'une telle ampleur justifie d'envisager des solutions mettant l'utilisateur au cœur du processus d'aide, non seulement au niveau de son exploitation, mais également au niveau de l'acquisition de connaissances. Par ailleurs, les utilisateurs rencontrent des problèmes extrêmement variés, liés d'une part à la méconnaissance du logiciel (il faut par exemple savoir qu'il existe un outil permettant de modéliser des coniques avant de pouvoir y arriver) et d'autre part aux objectifs fixés qui posent des problèmes purement conceptuels.

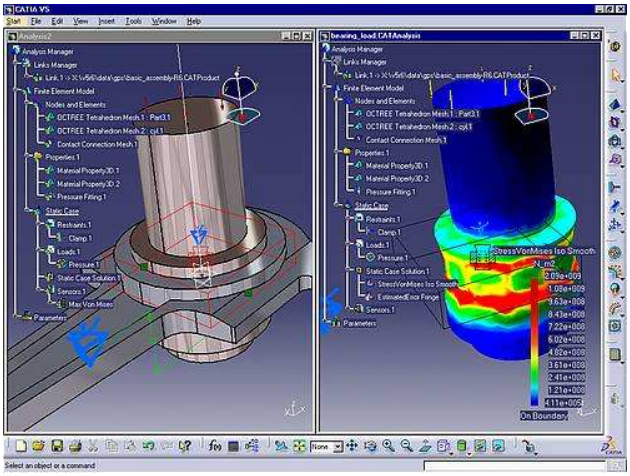


Figure 5 Illustration de l'environnement CFAO CATIA

5.2 Négociation de la prise en compte de l'expérience

Notre objectif est de réaliser un système d'aide générique, c'est-à-dire *facilitant* une tâche de l'utilisateur sans qu'elle soit explicitement décrite. Nous utilisons la théorie de la trace précédemment présentée pour y parvenir. Nous choisissons de considérer les événements (aspect dynamique) et les entités (regroupés dans des états successifs de l'espace de travail). Dans un tel système, un épisode est représenté par une suite d'événements, les relations entre événements étant gouvernées par les entités. La trace est donc gérée en deux parties : une partie statique composée des entités manipulées par l'utilisateur dans l'espace de travail, et une partie dynamique, composée des actions effectuées par l'utilisateur lors de son activité en inter-action avec l'espace de travail. Les actions effectuées, abstraites par le système pour permettre des comparaisons ultérieures, sont considérées comme *événements*. Les objets manipulés sont considérés comme *entités.*

Pour réutiliser l'expérience tracée à des fins de facilitation d'une tâche en cours, il faut donc atteindre deux objectifs : fournir une aide appropriée, en utilisant un mécanisme de calcul de similarité entre les épisodes mémorisés et la situation courante effective, et la fournir *à temps*, c'est-à-dire lorsqu'il y a le plus de chance que ce soit utile pour l'utilisateur, ce qui passe par un réglage effectué par l'utilisateur ainsi que par une qualification de son expertise.

La définition et l'implémentation d'un système de Raisonnement à Partir de l'Expérience Tracée (RàPET), nous permet d'espérer assurer la *pertinence* de l'aide fournie. Utiliser des expériences passées qui sont proches, structurellement et contextuellement, de la situation courante, devrait en effet permettre d'assurer à l'utilisateur une aide en rapport étroit avec le problème à résoudre. Le cœur d'un tel système repose sur la mesure de similarité entre la situation courante et les épisodes mémorisés.

5.3 Calcul de similarité entre l'historique et les épisodes mémorisés

Le calcul de similarité est la clef qui permet au système de fournir une aide pertinente. En effet, le fondement du RàPET est d'utiliser les expériences passées en vue d'assister la situation courante. Retrouver dans la base de connaissances quelle expérience mémorisée correspond le mieux à la situation actuelle est donc central.

Le calcul de similarité s'effectue en comparant *l'historique des événements courants* avec les séquences repérables dans les épisodes mémorisés comme illustré par la Figure 6. La similarité ainsi mesurée est double : similarité conceptuelle au niveau de l'événement (par exemple, on considérera les fonctions rechercher et rechercher/remplacer comme similaires, puisqu'elles découlent du même mécanisme) et similarité événementielle[7] au niveau de la séquence (la séquence ABEDH est similaire à ABDEH, par exemple).

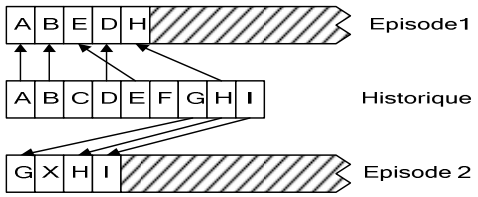


Figure 6 Calcul de similarité. Les lettres représentent différents événements, constitués à partir d'actions de l'utilisateur qui ont été abstraites. La partie hachurée des épisodes n'a pas encore été effectuée par l'utilisateur, et représente donc un potentiel d'aide.

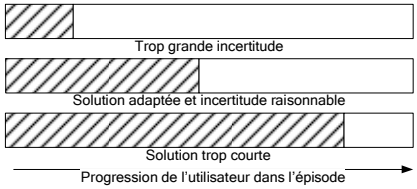

Trop grande incertitude
Solution adaptée et incertitude raisonnable
Solution trop courte
Progression de l'utilisateur dans l'épisode

Figure 7 Influence du début de l'épisode sur la fin de l'épisode. La partie hachurée représente la partie de l'épisode déjà effectuée par l'utilisateur, tandis que la partie claire représente l'aide apportée par le système, c'est à dire l'ensemble des activités à réaliser pour atteindre l'objectif supposé

Une autre caractéristique est la prise en compte de la longueur de la solution source (celle qui sera proposée). Comme on le voit dans la Figure 7, le calcul de similarité prend en compte le chemin restant à parcourir dans un épisode afin de déterminer sa pertinence. Ainsi, la similarité entre l'expérience retrouvée et la situation courante est assurée à la fois par une ressemblance dans les traces d'utilisation, mais également par l'estimation de l'efficacité

[7] Il s'agit d'une variante d'une similarité d'édition.

(nombre d'opérations à effectuer le plus faible possible) de la solution pour atteindre le but courant. En effet, il peut arriver qu'un épisode soit inclus dans un autre : par exemple, l'épisode « design de cercle » est inclus dans l'épisode « design de cylindre », puisqu'un cylindre est obtenu en extrudant un cercle. Dans ce cas, le choix de l'épisode dépend de sa portée, c'est à dire de la complexité des objets pour lesquels il est capable de faciliter la conception. Ainsi, dans cet exemple, l'épisode 'cylindre' a une portée supérieure à l'épisode 'cercle', puisque l'objet résultant cylindre est plus complexe que le cercle. L'assistant choisira donc un de ces épisodes, en prenant en compte le niveau d'expertise de l'utilisateur dans le cadre courant. Un utilisateur novice se verra proposer l'épisode 'cercle' alors qu'un utilisateur un peu plus expérimenté se verra proposer l'épisode 'cylindre', sa maîtrise justifiant un niveau d'abstraction un peu plus élevé.

Cette mesure de similarité, dans un monde idéal, doit être effectuée à chaque changement dans la trace courante. Il est en effet impossible de décréter où sera le début d'un épisode et il faut donc tous les tester. Dans la réalité, des contraintes matérielles interdisent une telle approche. Nous avons créé un système permettant de déléguer la mesure de similarité aux événements et entités eux-mêmes. Cela permet une parallélisation de ce calcul, ainsi qu'une réduction significative du nombre d'hypothèses à tester.

Afin de permettre cette délégation, nous avons du discriminer les objets sémantiquement intéressants pour l'utilisateur (car présentant une réalité conceptuelle) des objets informatiques tels qu'appréhendables par le système informatique. Ainsi, le modèle d'observation s'appuie sur quatre étages :

- Le premier étage représente la partie dynamique de la trace, c'est-à-dire les événements. Un événement n'a de sens que comme qualificatif dans l'activité conceptuelle, aussi doit-il être mis en relation avec les *réalisations* auxquelles il participe
- Les réalisations constituent le second étage. Une réalisation est un objet de conception au sens informatique du terme. Il s'agit donc, par exemple, d'une instance particulière de cercle possédant des paramètres (de rayon, d'épaisseur, …) précis et utilisé dans un cadre déterminé. Il est évident qu'une réalisation, contrainte par ces paramètres et le contexte de la séance, ne peut être utilisée comme représentant de connaissance. Il convient donc de l'abstraire dans une catégorie sémantiquement qualifiée, que nous appelons *entité*.
- L'entité est donc un élément de conception qui représente une catégorie de réalisation. L'entité cercle représente tous les cercles, et est donc reliée aux réalisations correspondantes. L'objectif ici est de permettre une mesure de similarité au niveau des éléments conceptuels. Cependant, le processus d'abstraction détruit les relations contextuelles entre objets. Ces relations doivent donc être reconstituées au sein d'*épisodes*
- L'épisode est la forme concrète reconnue grâce à une SiTeX. Il rassemble des entités mise en relations par des contraintes contextuelles. Le motif ainsi formé permet de reconstituer au vol les expériences passées en apportant deux contributions. La première constitue une abstraction de l'ordre imposé par la trace primitive. En effet, si la trace est séquentielle, la recherche de réalisations répondant aux contraintes contextuelles donne comme résultat un graphe orienté sans priorité d'un arc sur un autre. La seconde est l'intégration de plusieurs expériences à priori différentes au sein

du même épisode. S'il existe en effet des différences liées uniquement aux propriétés des objets de conceptions ou à la méthode utilisée pour concevoir ces objets (CATIA permet en effet de concevoir des objets de plusieurs manière différentes), celles-ci seront absorbées lors du processus d'abstraction menant de la trace à l'épisode.

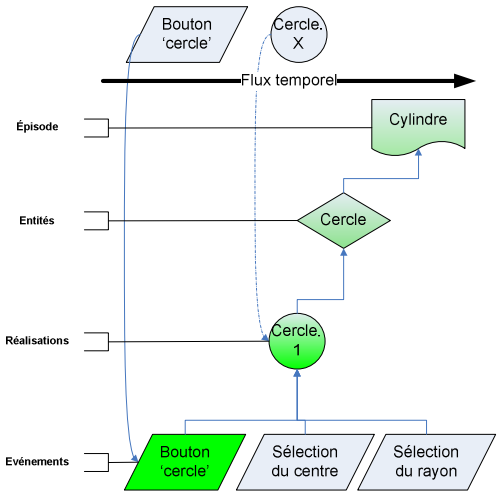


Figure 8 Représentation des épisodes et calcul de similarité

En utilisant cette représentation, le calcul de similarité passe par une transmission de message. L'événement, lorsqu'il est activé, c'est à dire lorsqu'un événement similaire survient au sein de la trace courante, envoie un message en direction des réalisations auxquelles il est rattaché. Celles-ci transmettent alors ce message à l'épisode via l'entité, seulement si une réalisation similaire existe au sein du document de travail. L'épisode vérifie alors la cohérence du message avec ceux qu'il a déjà reçus, notamment au niveau des relations entre entités et, si cette cohérence est assurée, augmente son niveau de similarité avec la situation courante. Le système classe alors les épisodes par similarité, et détermine ensuite lesquels sont à présenter à l'utilisateur.

Une telle approche fondée sur la réutilisation de l'expérience pour suggérer des suites possibles dans un contexte bien précis répond à l'exigence de mise en situation de l'aide, mais il reste à déterminer à quel moment le système d'aide doit intervenir. En effet, un tel système est appelé à intervenir de manière spontanée, ce qui ne lui donne pas le droit à l'erreur et il sera rapidement disqualifié s'il ne rend que rarement service car se « trompant » de situation ou intervenant de manière inopportune. Comme on peut le comprendre dans la Figure 7, si l'intervention intervient trop tard, la fin de l'épisode sera trop courte pour être d'une quelconque utilité et, par exemple, il est en général inutile d'intervenir pour dire simplement « appuyez sur OK ». Par ailleurs, une intervention trop rapide risque fort d'être peu pertinente. En effet, la similarité entre la trace primitive et l'épisode est alors peu significative et énormément d'épisodes seront sélectionnables. Or, dans le cas d'une intervention spontanée, une intervention importune conduit souvent l'utilisateur à rejeter l'aide et ainsi, un système d'aide qui demanderait à l'utilisateur si celui-ci veut écrire un courrier dès qu'il tape le mot « cher » serait rapidement rejeté. Pour nous approcher d'une intervention « au bon moment » nous proposons de compléter le dispo-

sitif de réutilisation de l'expérience tracée par un mécanisme original de détermination interactive du seuil d'intervention.

5.4 Détermination du seuil d'intervention

Au delà des solutions instrumentées, exploitant les travaux sur le fonctionnement de la mémoire humaine (Card, et al., 1983; Damas, 2003) et qui permettent de borner temporellement ou sur un nombre d'événements, une estimation d'un seuil d'intervention adapté doit se fonder sur une interaction avec l'utilisateur.

En effet, en s'appuyant sur le phénomène d'hystérésis et sur celui d'écho mémoriel, il serait envisageable de fixer un seuil d'intervention fixe au bout de X interactions par exemple. Un tel seuil, cependant, ne prend pas en compte l'utilisateur singulier, mais un utilisateur moyen. Une solution classique est de catégoriser les utilisateurs, le seuil d'intervention étant différent, par exemple, pour les novices et les experts. Cela peut conduire rapidement à un grand nombre de catégories, par niveau, par domaine d'expertise, etc. et demande à l'utilisateur de gros efforts pour catégoriser son travail de manière à obtenir l'ajustement du seuil d'intervention adéquat

Notre approche consiste à permettre à l'utilisateur de régler directement le seuil d'intervention, par des mécanismes relatifs à la complexité de sa tâche mais restant intuitifs et explicites. Nous proposons d'exprimer un seuil d'intervention non plus en valeur absolue de longueur de séquence (X interactions) mais en valeur relative (X %) à la *complexité* d'un épisode.

La complexité d'un épisode est calculée en fonction de plusieurs paramètres. Il convient tout d'abord de considérer l'épisode au sein du corpus d'épisodes constitué. Certains épisodes sont en effets contenus dans d'autres (par exemple, la conception de cercle est inscrite dans la conception du cylindre). Cette mesure est faite de manière statique et rafraichie régulièrement. Une autre composante de cette mesure concerne la concrétisation de l'épisode. Cette concrétisation est le processus inverse de celui de construction de l'épisode à partir de la trace. Il consiste à recréer une séquence événementielle permettant de retrouver l'épisode par le processus d'abstraction. Les événements pris en compte ici sont considérés comme étant tous de même niveau sémantique. En effet, ces événements sont ceux collectés par les agents d'observation, qui sont limités de facto à ce qui est visible depuis le logiciel aidé. Leur sémantique est donc très faible, puisque ces événements sont liés à des éléments d'interface précis (clic sur un bouton, par exemple). C'est pourquoi nous les considérons sur un pied d'égalité. Cette recréation se fait en situation, c'est-à-dire en relation directe avec le travail actuel de l'utilisateur. Cela fait donc de la complexité de l'épisode une mesure composée d'un élément statique et d'un élément dynamique. Le seuil d'intervention du système est exprimé comme relatif à la complexité des épisodes pressentis. La Figure 9 exprime ce que donnerait un seuil d'intervention constant dans notre système.

A l'aide d'une interface minimale comportant deux boutons '+' et '-' situés à coté des épisodes proposés par l'assistant, l'utilisateur peut ajuster le seuil d'intervention pour une classe d'épisodes à travers un épisode précis (voir l'illustration du Tableau 1. Nous plaçons donc un point dans notre espace {complexité, seuil} qui va servir de point de gravité pour la courbe seuil=f(complexité). Ici, l'utilisateur ne connaît pas la complexité de l'épisode pour lequel il propose un réglage. Il indique simplement son opinion sur le

moment de la présentation de cet épisode, qui est ensuite traduit par le système sous la forme d'un rapport entre la complexité de l'épisode et le pourcentage de cet épisode déjà effectué. Cela donne une courbe représentée en Figure 10. Ce système offre ainsi à l'utilisateur un moyen de régler le système d'aide de manière très simple, puisqu'un nombre très réduit d'interventions sont requises pour obtenir une courbe répondant à ses besoins.

Dans ce système, une classe d'épisode est représentée par une complexité donnée. Ainsi, tous les épisodes de même complexité appartiennent à la même classe. Le réglage effectué pour un épisode affecte donc tous ceux de complexités similaire et voisine, à travers l'ajustement de la courbe Ainsi, les réglages de l'utilisateur sont indépendants de la sémantique des épisodes. Or, celle-ci est très importante dans le choix du moment d'intervention, car elle est liée à la sémantique de l'objectif atteint au travers de l'épisode. Afin de prendre en compte cette sémantique, il convient de modéliser l'expertise de l'utilisateur, de manière à ajuster le moment d'intervention non seulement à ses attentes mais également à cette expertise.

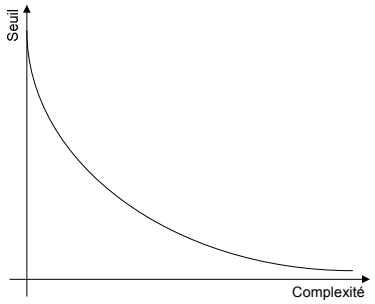


Figure 9 Seuil d'intervention constant. Un seuil d'intervention constant en nombre d'actions se traduit par une courbe hyperbolique lorsqu'on l'exprime en pourcentage de la complexité des épisodes. En effet, si 5 actions représentent 100% d'un épisode de complexité 5, cela ne fait que 10 % d'un épisode de complexité 50

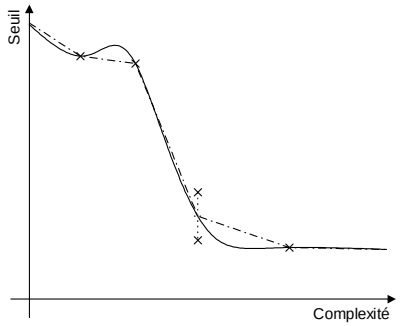


Figure 10 Seuil d'intervention réglé par l'utilisateur. Les points représentent les ajustements faits par l'utilisateur sur des épisodes d'une complexité donnée. La courbe ainsi obtenue définit son profil. Pour des raisons pratiques, elle est transformée en ligne brisée (en pointillés) par le système.

5.5 Modéliser l'expertise de l'utilisateur

Pour un utilisateur expert, une aide sur des épisodes peu complexes ne représente aucun intérêt. Il convient donc de réduire l'espace de recherche dans

la base de cas en tenant compte de l'expertise de l'utilisateur. Cependant, sur des suites logicielles, un utilisateur peut être expert dans une partie de la suite et totalement novice dans une autre. On peut par exemple maîtriser Word pour des courriers, même complexes avec des fusions par exemple, et être incapable de mettre en forme correctement un article scientifique, même simple, avec ce même outil. Modéliser l'expertise de l'utilisateur ne peut donc pas se faire globalement, mais par *domaine*.

Un domaine est représenté par un ensemble d'épisodes de la base, et est créé soit implicitement en fonction d'informations de contexte (« tous les épisodes de conception de courriers » ou « tous les épisodes de conception d'articles scientifiques »), soit explicitement selon des critères donnés par un administrateur. La création explicite de domaine par un administrateur correspond à des contraintes liées aux domaines d'interventions des utilisateurs potentiels du système. La création implicite, elle, est vue du point de vue du système, de manière à pouvoir qualifier l'expertise de l'utilisateur dans des outils qui sont proches d'un point de vue conceptuel. Les informations de contexte utilisées sont prises dans le système aidé et donc définie au moment de la conception de l'agent facilitateur (une évolution envisageable est de permettre à l'utilisateur de définir ses propres critères contextuels)

Il s'agit alors d'estimer le plus automatiquement possible l'expertise dans un domaine. Cette estimation est dans notre cas dépendante de la complexité des épisodes que l'utilisateur réalise avec ou sans aide versus la complexité des épisodes au cours desquels l'utilisateur sollicite l'aide. Le principe consiste à repérer quel est l'épisode le plus complexe n'ayant pas nécessité d'aide. Ce niveau de complexité va déterminer un niveau d'expertise considéré dans les domaines de classement de cet épisode.

Cette information sur le niveau d'expertise permet de restreindre l'espace de recherche dans la base d'épisodes. Les cas à la complexité révélant une expertise sensiblement inférieure à l'expertise estimée de l'utilisateur courant ne seront pas proposés spontanément. Cela poursuit un double objectif : éviter d'importuner l'utilisateur avec des cas trop simples pour son niveau et améliorer l'efficacité du système en réduisant le champ des possibles et donc les possibilités d'erreur.

L'évaluation du domaine de travail courant se fait à travers deux mécanismes. Le premier concerne les habitudes de travail de l'utilisateur. Les domaines sont pondérés selon le nombre de cas résolus par l'utilisateur dans chacun d'eux. Ainsi, le système sera plus enclin à proposer de l'aide sur des domaines où l'utilisateur à l'habitude de travailler, épargnant ainsi des propositions d'aide parasites. Le second est lié aux informations contextuelles de la session courante. Un domaine étant créé en fonction d'informations de contexte, il est possible de mettre en adéquation celles-ci avec le contexte courant, de manière à donner une indication du domaine dans lequel se situe l'utilisateur. Le domaine du courrier sera caractérisé par l'utilisation de l'outil de fusion, par exemple, tandis que celui de l'article scientifique le sera par l'insertion de références bibliographiques.

5.6 Illustration sur un scénario d'usage de l'outil développé

La technologie de détermination du seuil d'intervention présentée ici a été implémentée au sein d'un prototype d'assistant pour CATIA. L'interface

d'interaction pour faciliter l'appropriation du système d'aide est illustrée dans le Tableau 1.

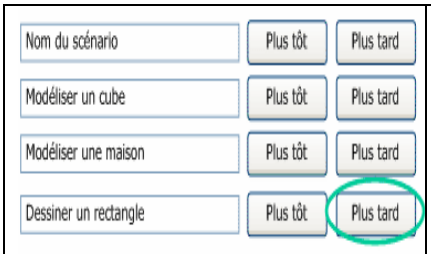
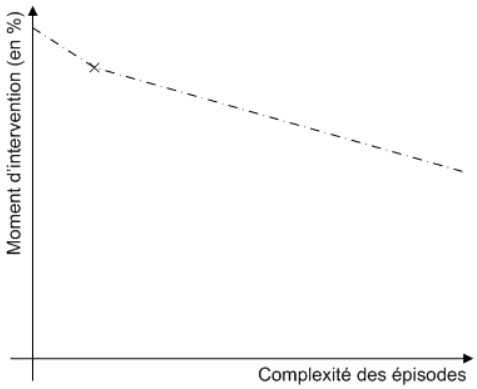
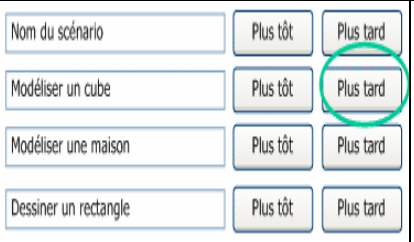
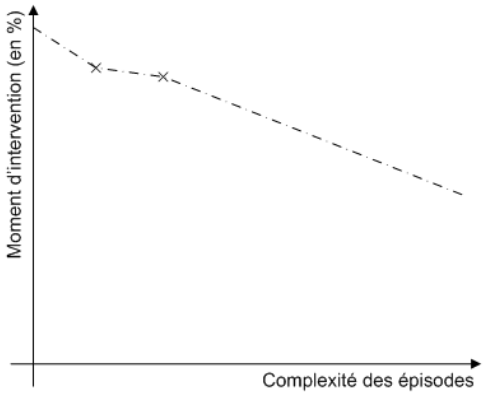
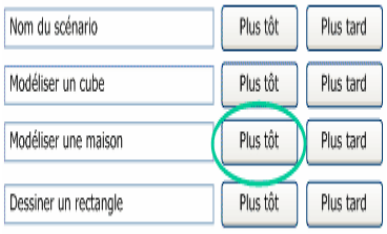
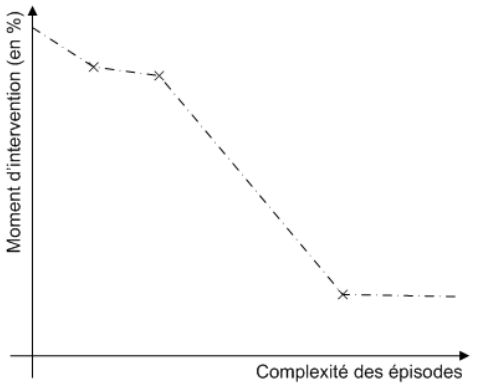
 *L'utilisateur indique que le compagnon doit retarder sa proposition d'aide pour les épisodes de type « dessiner un rectangle » considéré comme peu complexe par le système.*	
 *L'utilisateur demande au compagnon de retarder sa proposition d'aide pour les épisodes de type « modéliser un cube », pourtant un peu plus complexe que « dessiner un rectangle ».*	
 *L'utilisateur demande au compagnon une aide précoce dans le cas d'épisodes de type « Modéliser une maison » considéré comme complexes par le système (et donc probablement par l'utilisateur lui-même).*	

Tableau 1 Personnalisation de l'aide apportée par le compagnon CATIA

Les premiers tests ont eu lieu, faits par 3 utilisateurs sur une base d'environ 500 éléments et environ 15 épisodes. Les trois utilisateurs ainsi observés pré-

sentent des compétences très différentes. Les profils obtenus à l'issue de ces tests sont présentés en Figure 11. Les trois courbes obtenues, montrent trois approches de l'aide avec en abscisse la complexité et en ordonnée le seuil d'intervention exprimé en pourcentage de la complexité.

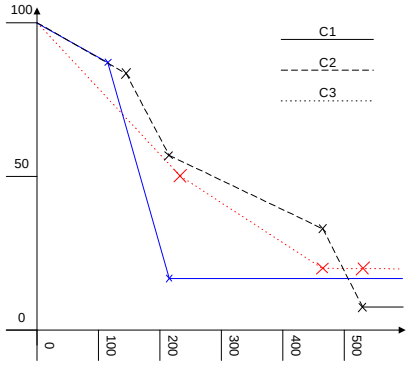


Figure 11 Résultats avec seuil exprimé en pourcentage

La courbe C1, réalisée par un expert sur quelques domaines (développeur expérimenté sur CATIA), montre un rejet de l'aide pour des épisodes très simples. Ensuite, l'aide attendue reste sur un seuil constant, indiquant que l'utilisateur souhaite de l'aide une fois qu'une certaine discrimination a été effectuée. La différence entre ces deux phases reflète les attentes de l'utilisateur, qui ne veut être aidé que sur des sujets relativement complexes.

La courbe C2, réalisée par un expert dans de nombreux domaines (utilisateur confirmé en milieu industriel), indique que l'utilisateur attend de l'aide plus tardivement que celui qui a réalisé C1, hormis sur les épisodes simples. Cette courbe qui décroît régulièrement reflète une attente d'aide relativement indépendante de la complexité, mais constante en termes temporels, puisqu'elle se rapproche de la courbe de la Figure 9, qui est celle d'un seuil d'intervention constant.

La courbe C3, réalisée par un utilisateur novice (n'ayant jamais travaillé sur un outil de conception 3D), reflète une attente d'aide sur des épisodes complexes plutôt que sur des épisodes simples. L'utilisateur, ici, préfère apprendre par lui même, avec la méthode des essais et erreurs, mais attend du compagnon qu'il lui fixe un objectif plus large à atteindre, de manière à créer un cadre dans lequel réaliser ses expériences.

Les courbes C1 et C3, bien que relativement similaires, sont en fait la traduction d'attentes très différentes, alors que ces deux utilisateurs auraient pu être rattachés au même groupe dans le cadre d'un seuil fixé par catégorie d'utilisateur. Sur un très petit nombre d'utilisateurs, nous observons des attentes différentes, qui se traduisent par des courbes différentes. Les utilisateurs ont donc une approche différente et des attentes personnalisées vis-à-vis de l'aide. Nous pensons donc qu'une approche individualisée telle que celle présentée ici permet d'améliorer l'aide par rapport à des méthodes catégorielles ou péremptoires.

6. DISCUSSION

La présentation de la mise en œuvre d'un facilitateur fondé sur les principes de l'exploitation de l'expérience tracée en intégrant des mécanismes de *négociation* de la manière d'exploiter l'expérience montre qu'il est possible de considérer l'utilisateur comme *acteur* de sa propre appropriation. Le fait que la négociation se fasse en prenant comme base des épisodes repérés comme pertinents par l'utilisateur et support d'estimation concret d'un niveau de complexité rend *assez triviale* la chose et démontre que le couplage environnement-utilisateur n'est pas une vue de l'esprit. Dans cette implémentation, la présentation de la trace à l'utilisateur n'a pas été étudiée en tant que telle[8] et seul l'effort d'adaptation est pris en compte au travers de la mesure de similarité qui est l'objet implicite des interactions pour « régler » l'aide. Elle n'apparaît que par les épisodes candidats pour assister et qui sont sans cesse remis en question au fur et à mesure que l'activité se prolonge, ce qui explique l'importance d'un réglage du délai d'intrusion du compagnon dans la tâche en cours. D'une manière générale, cet *objet informatique* nouveau constitué par la trace d'utilisation, pose la question de sa représentation à l'utilisateur. Cette représentation doit être *négociée* avec l'utilisateur pour qu'il puisse se l'approprier à des fins de réutilisation comme de partage. On peut alors voir cette question comme l'émergence d'un langage permettant de *parler* de la trace d'inter-actions. Nous tentons d'apporter des réponses à cette question en nous inspirant de l'émergence de langage à la manière des travaux de Luc Steels (Stuber, et al., 2005). L'implémentation qui a été présentée a été possible parce qu'un éditeur de suite logicielle, ayant tous les moyens de collecter les événements en instrumentalisant ses sources, a exploité cette maîtrise pour réaliser l'agent d'observation chargé de la collecte. Le modèle d'utilisation exploité pour cette réalisation serait, selon notre définition, le premier modèle, très utile au moment du déploiement de l'environnement, mais devant être évolutif en fonction du point de vue *d'un métier* par exemple. C'est en effet la seule façon de rendre compte de l'environnement informatique tel qu'il est *perçu* par l'utilisateur. Nous pensons qu'il est par ailleurs de plus en plus difficile pour les concepteurs d'imaginer cet environnement, compte tenu du fait qu'ils n'en sont concepteurs que d'une partie, tout en considérant *de leur point de vue* les autres parties comme inter-opérantes. C'est ainsi qu'ils prendront la précaution d'appareiller l'exportation de leurs structures de données pour qu'elles puissent être exploitées par d'autres logiciels. C'est particulièrement vrai pour tout ce qui est visualisable pour l'utilisateur, qu'il aura tendance à vouloir *copier-coller* d'une suite logicielle à une autre. Comme on le comprend rapidement, si l'on souhaite vraiment intégrer les traces comme objets informatiques en tant que tels avec leurs outils de gestion et d'exploitation, il est nécessaire de définir ce que serait un *système à base de traces*. Ce point est discuté dans (Mille et Prié, 2006). Si nous considérons avec beaucoup d'enthousiasme les possibilités offertes par l'exploitation de traces d'utilisation comme inscriptions de connaissances potentielles pour l'utilisateur lui-même et ceux avec qui il interagit, nous sommes également impressionnés par l'étendue du travail à réaliser pour valider ce principe aussi bien au niveau expérimental qu'au niveau d'une offre nouvelle dans les environnements informatiques. Pour appro-

[8] La société Dassault Systèmes cherchait à développer un système d'aide induisant des schèmes d'utilisation et ne souhaitait pas représenter la trace en train de se construire, se concentrant sur la présentation d'épisodes susceptibles d'aider dans le contexte tracé.

fondir les concepts en les confrontant aux théories que nous considérons comme résonnant avec eux, il est plus que jamais nécessaire de considérer l'interdisciplinarité comme condition *sine qua non* de nos travaux en cours et à venir. Une « théorie de la trace » est actuellement en cours de formalisation grâce à un tel travail interdisciplinaire[910].

7. CONCLUSION

Nous avons considéré dans cet article la possibilité concrète de mettre en place les conditions d'un couplage utilisateur-environnement informatique explicite pour l'utilisateur et permettant une *aide* singulière façonnée par lui. Cette réalisation a été rendue possible par le principe d'utilisation de l'expérience tracée comme inscription de connaissances re-mobilisables en situation dans des conditions *négociées* avec l'environnement. Dans cette illustration, la négociation porte d'une part sur le délai d'intrusion du compagnon dans l'activité lorsqu'un épisode tracé semble intéressant comme facilitateur pour un épisode de conception nouveau et d'autre part sur le niveau de complexité pour lequel l'utilisateur entend se faire assister par le compagnon. L'important est de noter le caractère *trivial* de l'interface de négociation qui permet pourtant à l'utilisateur d'arriver à ses fins sans représentation explicite ni du délai, ni de la complexité. Pour concevoir de tels systèmes fondés sur l'exploitation des traces d'interaction pour offrir une aide « en situation », il est nécessaire d'intégrer dans le cycle de conception de l'outil objet de l'aide des phases de modélisation de l'utilisation en cours d'activité. L'outil lui-même doit être construit pour faciliter cette modélisation par l'utilisateur lui-même. Dans le cas du compagnon CATIA, l'usage réflexif des traces par des praticiens experts a soutenu cette activité de modélisation, permettant de fournir au compagnon développé les éléments de description des interactions et des signatures expliquées de tâches (issues de l'expérience). Les recherches qui nous semblent actuellement prioritaires concernent : l'analyse de l'impact possible de l'intégration *d'objets traces* dans les environnements informatiques, en particulier avec la notion de *système à base de traces* voire de *systèmes d'exploitation intégrant l'objet trace*. L'étude du potentiel de la propriété de réflexivité des traces pour l'appropriation et donc *l'instrumentation* d'un environnement technique.

Nous retenons, en référence à Simondon (2005, pp. 88-89) l'idée selon laquelle *« L'instrument équipe le système sensoriel, il sert à prélever de l'information, tandis que l'outil sert à exercer une action »,* idée reprise par Bruillard (1998) posant que *l'outil façonne et l'instrument instruit.* Un environnement informatique est un ensemble impressionnant d'outils qui peuvent se révéler des instruments, c'est-à-dire qui peuvent servir à comprendre, à s'approprier les choses, mais ne sont pas, en général, prévus pour cela. Les outils les plus puissants sont les plus difficiles à utiliser comme instruments et c'est pourquoi nous proposons que la trace d'utilisation puisse être un médium facilitant la genèse instrumentale des outils d'un environnement informatique et, par suite, puisse devenir un puissant moyen d'appropriation de l'informatique par ses utilisateurs.

---

[9] http://theorie-trace.liris.cnrs.fr/

[10] http://praxis.inrp.fr/praxis/projets/projets/acteurs

REFERENCES

Aamodt, A., Plaza, E. (1994). Case-Based Reasoning: Foundational Issues, Methodologic Variations and System Approaches. *AI Communications* Vol 7: 39-59.

Bachimont, B. (2004). *Pourquoi n'y-a-t-il pas d'expérience en ingénierie des connaissances*. In Actes de IC'2004, LYON.

Brassac, C. (2006). Computers and Knowledge: A Dialogical Approach. *AI & Society* Vol 20/3 (February 2006), 1-22.

Card, S. K., Morgan, T. P., Newell, A. P. (1983). *The Psychology of Human-Computer Interaction*. Hillsdale, NJ: Erlbaum.

Champin, P.-A. (2003). *Ardeco: An Assistant for Experience Reuse in Computer Aided Design*. In Workshop From Structured Cases to Unstructured Problem Solving Episodes - WS 5 of ICCBR'03, Trondheim (NO), NTNU.

Champin, P.-A., Mille, A., Prié, Y. (2003). *MUSETTE: Modelling USEs and Tasks for Tracing Experience*. In Workshop From Structured Cases to Unstructured Problem Solving Episodes - WS 5 of ICCBR'03, Trondheim (NO), NTNU.

Charlet, J. (2004). *L'ingénierie des connaissances, entre science de l'information et science de la gestion*. Rapport de recherche SIC 805, http://archivesic.ccsd.cnrs.fr.

Clot, Y. (1999). *Avec Vygotski*. Paris : La Dispute.

Corvaisier, F., Mille, A., Pinon, J.-M. (1997). *Information Retrieval on the WWW Using a Decision Making System*. In Proceedings: RIAO'97.

Corvaisier, F., Mille, A., Pinon, J.-M. (2000). Recherche assistée de documents indexés sur l'expérience : mesure de similarité des épisodes de recherches sur le Web. In J. Charlet, M. Zacklad, G. Kassel and D. Bourigault (éds), *Ingénierie des Connaissances: évolutions récentes et nouveaux défis* (pp. 387-406). Paris: Eyrolles.

Damas, L. (2003). *Étude théorique et pratique de la production d'effets d'amorçage de la mémoire*. Thèse de doctorat soutenue à Lyon, France, UCBL.

Dewey, J. (1934). *Art as Experience*. New York: Perigee Books.

Héraud, J.-M., Mille, A. (2003). *Pixed: assister l'apprentissage à distance par la réutilisation de l'expérience*. In Proceedings de l'Atelier Raisonnement à Partir de Cas, Plateforme AFIA'03 Laval.

Herbeaux, O., Mille, A. (1999). ACCELERE: A Case-based Design Assistant for Closed Cell Rubber Industry. *Knowledge Based Systems*, Vol 12 (1999), 231-38.

Laflaquière, J., Ciaccia, A. (2005). Facilitation de tâches informatiquement médiées: une approche centrée sur la réflexivité de l'utilisation. In *6ème colloque des jeunes chercheurs en sciences cognitives*, Bordeaux, pp. 117--17.

Leontiev, A. N. (1984). *Activité, conscience, personnalité*. Moscou: Éditions du Progrès.

Mille, A., Fuchs, B., Chiron, B. (1999). Raisonnement fondé sur l'expérience: un nouveau paradigme en supervision industrielle? *Revue d'Intelligence Artificielle* Vol 13, 97-128.

Mille, A., Prié, Y. (2006). *Une théorie de la trace informatique pour faciliter l'adaptation dans la confrontation logique d'utilisation / logique de conception*. In 13èmes journées de Rochebrune: Rencontres interdisciplinaires sur les systèmes complexes naturels et artificiels, Rochebrune, Megève, ENST.

Rabardel, P. (1995). *Les hommes et les techniques, une approche cognitive des instruments contemporains*. Paris: Armand Colin.

Riesbeck, C. K., Schank, R. (1989). *Inside Case-based Reasoning*. Hillsdale, NJ: Lawrence Erlbraum.

Schank, R. C. (1982). *Dynamic Memory: A Theory of Reminding and Learning in Computers and People*. Cambridge University Press.

Stiegler, B. (2005). Désir et connaissance. Le mort saisi par le vif. Eléments pour une organologie de la libido. *Revue d'intelligence artificielle: Numéro spécial* Vol 19/1-2, 13-29.

Stuber, A., Hassas, S., Mille, A. (2005). *Languages Games for Meaning Negotiating Between Human and Computer Agents*. In ESAW 2005, Engineering Societies in the Agent's World, Kusadasi, Aydin, Turkey, Springer Verlag.